

Bridging Lake and Database: Bringing Real-time to Iceberg

Enabling real-time analytics on massive lake data

Speaker

Owen Xiao and Kevin Shen

Owen Xiao

Apache Doris PMC Member & VeloDB co-founder

<https://www.linkedin.com/in/owen-xiao/>

Kevin Shen

Senior Product Manager at VeloDB,

<https://www.linkedin.com/in/yuankai-kevin-shen/>



Agenda:

1. Why bring real-time workloads matters more than ever
2. Key Challenges of Tackling Real-Time Workloads on a Lakehouse
3. Brief Intro to Apache Doris and Apache Iceberg
4. Apache Doris + Iceberg for High Performance analytics on lakehouse
5. Beyond Apache Doris
6. Roadmap
7. Community and Commercial Support

From customer-facing to agent-facing analytics

Why Real-Time Matters

Instant Customer Insights



- Capture customer behavior immediately
- Deliver personalized recommendations on the fly
- Boost conversion and satisfaction

Empowering Agents with Live Context



- Give agents up-to-date customer info
- Support faster and more accurate decisions
- Improve issue resolution efficiency

Proactive Operations & Issue Resolution



- Detect potential issues before they escalate
- Trigger real-time alerts for anomalies
- Enable rapid intervention and risk reduction

Key Challenges for Real-Time Workloads on a Lakehouse

Achieving real-time analytics on data lakes presents several significant technical hurdles that must be overcome to deliver database-like performance on lake storage:



High Concurrency & Low Latency

Supporting hundreds of concurrent users with sub-second query response times on petabyte-scale data lakes requires specialized query engines and optimizations.



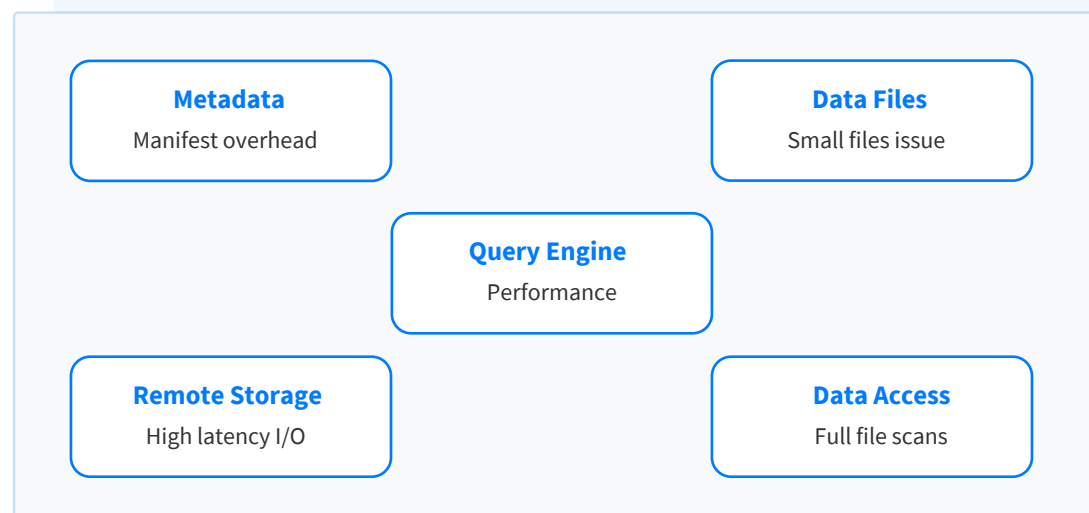
Remote I/O Performance

Data lake storage (S3, HDFS) has higher latency compared to local storage. Overcoming this requires intelligent prefetching, parallel I/O, and multi-level caching strategies.



Efficient Data Scanning

Small files problem, manifest handling, lack of efficient indexing, and suboptimal sort orders can severely impact query performance on Iceberg tables.



Iceberg's Challenge:

While Iceberg provides ACID transactions and metadata management, real-time query performance still requires extensive optimization at every layer of the query pipeline.

Apache Doris Quick Intro

Apache Doris is a modern, high-performance MPP (Massively Parallel Processing) database system designed specifically for analytics workloads. It combines speed, scalability, and simplicity to deliver fast analytics on large datasets.



Modern MPP Architecture:

Column-oriented storage engine with vectorized execution and distributed query processing optimized for analytical workloads.

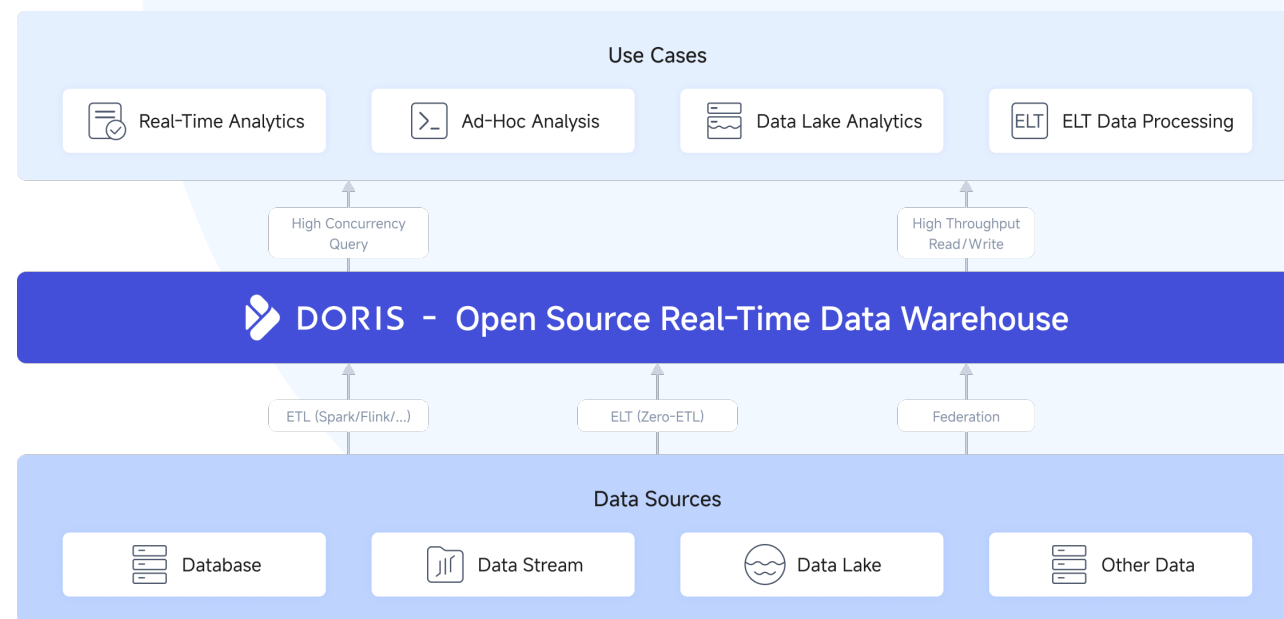


Unified Analytics Engine:

Supports multiple data models (star schema, flat tables, pre-aggregation) and diverse workloads from batch processing to real-time analytics.

Key Capabilities:

Sub-second queries on petabyte-scale data, real-time data ingestion, high concurrency support, and MySQL-compatible protocol for seamless integration with BI tools.



Apache Doris Community & Ecosystem

The Apache Doris community has seen extraordinary growth since becoming a Top-Level Apache project in 2022. With a focus on both open source collaboration and enterprise readiness, Doris has built a vibrant ecosystem spanning multiple industries and use cases.

700+

Contributors

14,000+

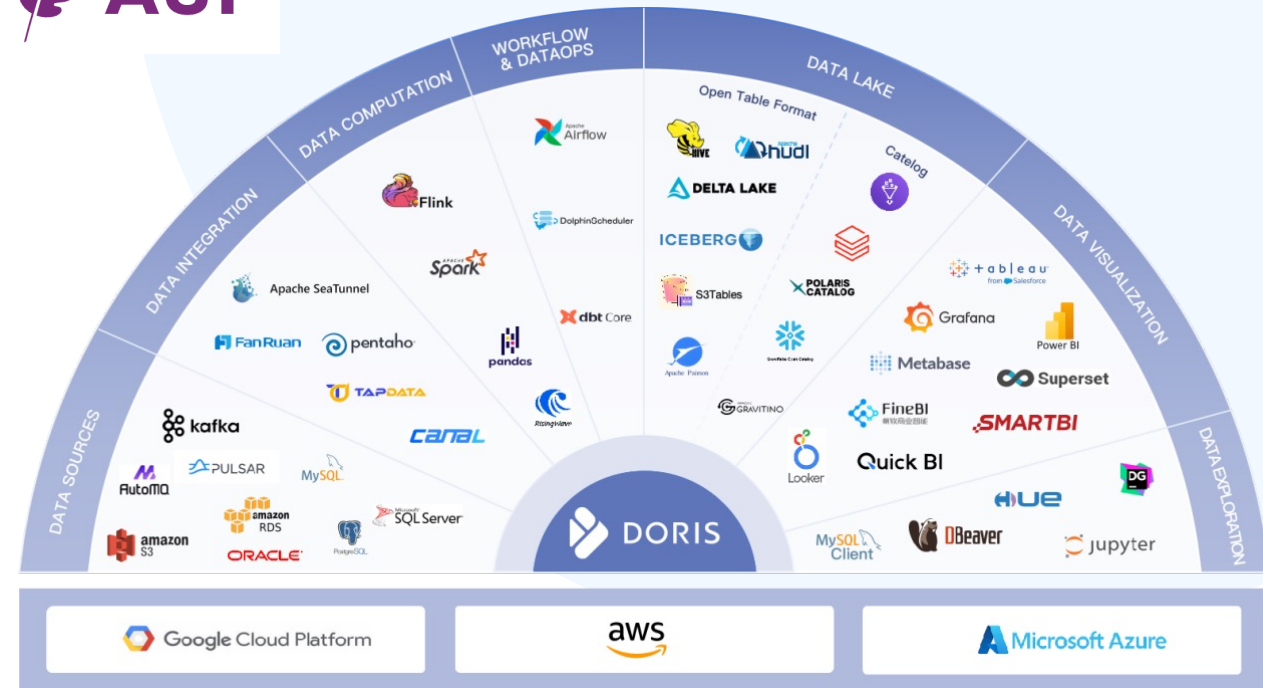
GitHub Stars

5,000+

Enterprise Users

Industry Adoption:

Widely deployed across finance, e-commerce, internet services, manufacturing, telecommunications, and IoT sectors for real-time analytics workloads.



Apache Iceberg

Apache Iceberg is an open table format designed for large scale analytic analytics. It brings the reliability and simplicity of SQL tables to the big data ecosystem such as lakehouse, while making it possible for engines like Spark, Trino, Flink, and Apache Doris to safely work with the same tables concurrently.



ACID Transactions:

Iceberg provides atomic, consistent, isolated, and durable transactions that ensure data reliability even with concurrent reads and writes.



Schema Evolution:

Seamlessly add, drop, rename, or reorder fields in tables without affecting query performance or causing conflicts between concurrent reads and writes.



Time Travel:

Query data as it existed at a specific point in time or at a specific snapshot, enabling reproducible queries, auditing, and simplified data recovery.

Table Metadata Layer

- Schema definitions, partition specs, and sorting orders

Snapshot & Manifest Layer

- ACID transactions, time travel capabilities

Data Files Layer

- Optimized file formats like Parquet, ORC, Avro

Format Evolution

Backward-compatible upgrades

Hidden Partitioning

Partition pruning without partition in query

Branch & Tag

Git-like versioning for data

High Performance analytics on Lakehouse with **Apache Doris** and **Iceberg**

The Lakehouse Architecture Based on Apache Doris

Apache Doris integrates seamlessly with Iceberg to create a powerful lakehouse architecture that combines the flexibility of data lakes with the performance of analytical databases.



Native Integration:

Doris provides native connectors to Iceberg tables, allowing for direct querying without data movement or duplication.



Accelerated Analytics:

While preserving Iceberg's data governance capabilities, Doris adds MPP query acceleration and optimization for real-time analytics.

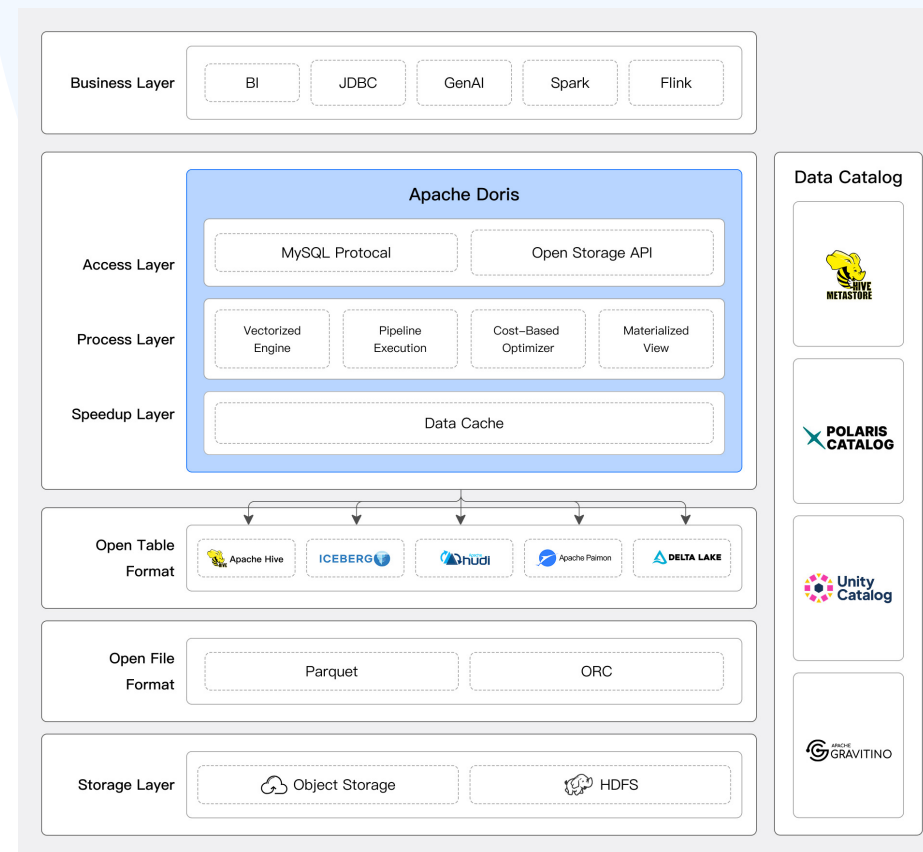
Key Usage Scenarios

Real-time dashboards on large-scale historical data

Unified batch and streaming analytics

Data lake acceleration without ETL

Federated query across multiple data sources



Doris Real-time Queries on Iceberg(1)

IO Optimization

Apache Doris achieves high-performance real-time queries on Iceberg tables through a series of sophisticated engine and I/O optimizations designed specifically for lake data processing.



Prefetch & Merge IO:

Intelligent data prefetching mechanisms with adaptive read-ahead strategies reduce I/O wait times by 60-80%. Batch read requests are merged to minimize network round trips.

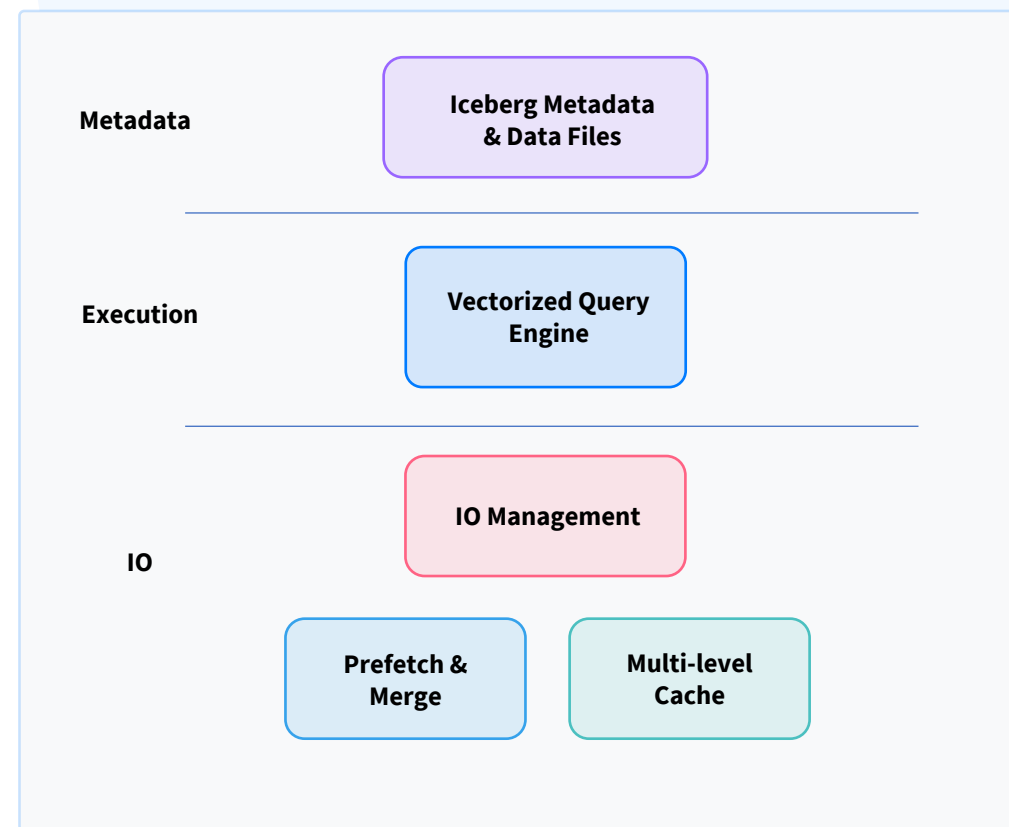


Multi-level Caching:

Local SSD cache for hot data blocks, distributed metadata cache for Iceberg manifests, and adaptive memory buffer management maximize cache hit rates.

Iceberg Metadata Optimization:

Native integration with Iceberg's metadata layer for efficient partition pruning, statistics utilization, and snapshot selection, reducing data scan volumes by up to 95% for typical analytical queries.



Doris Real-time Queries on Iceberg(2)

Schedule & Scan

Achieving real-time query performance on Iceberg tables requires sophisticated scheduling and scanning optimizations. Apache Doris implements several advanced techniques to maximize efficiency:



Priority Scan Scheduler:

Intelligently prioritizes scan tasks based on data locality, task complexity, and cluster load balance to minimize query latency.



Runtime Partition Prune:

Eliminates unnecessary partition scans at runtime using predicate conditions, dramatically reducing data scan volume.

Performance Impact:

These optimizations collectively reduce query latency by 70-90% compared to standard Iceberg scans, delivering near-database speed for data lake queries with **predictable latency**.



Doris Real-time Queries on Iceberg(3)

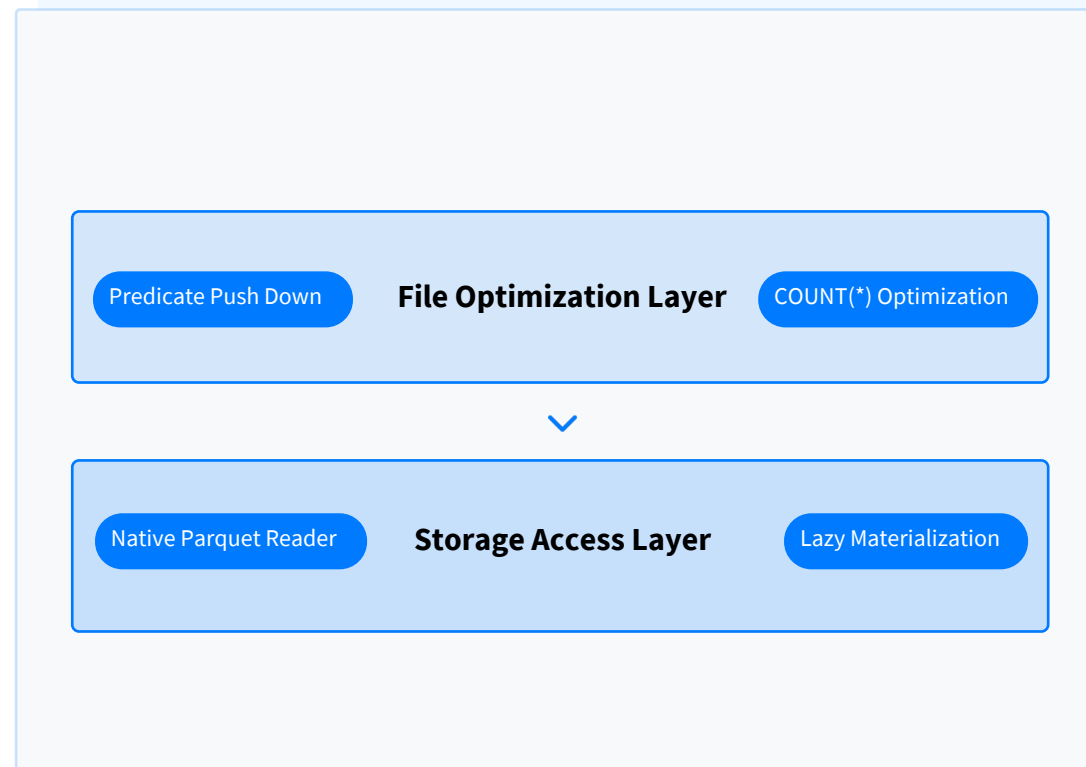
File Read Optimization

Apache Doris implements several key optimizations for efficient file reading from Iceberg tables, dramatically improving query performance on data lakes:

- 
Native Parquet Reader
 Purpose-built reader optimized for vectorized processing, bypassing generic interfaces for 3-5x faster file access.
- 
Predicate Push Down
 Filters pushed into storage layer, eliminating unnecessary data reads. Leverages Iceberg metadata for precise filtering.
- 
Lazy Materialization
 Defers column loading until actually needed, reducing memory usage and I/O by 40-60% for wide tables.

Performance Impact:

These optimizations reduce data scan volume by up to 95% and cut query latency by 5-10x compared to traditional data lake queries.



Materialized View: The Bridge Between Lake and Database

Materialized views are pre-computed, persistent result sets that transform complex analytical queries into simple table lookups, significantly accelerating query performance while maintaining data freshness.



Materialized Storage:

Apache Doris stores pre-computed aggregates, joins, and filtered datasets in its highly optimized columnar format, dramatically reducing query processing time.



Transparent Query Rewrite:

Queries automatically leverage materialized views without requiring any changes to application code, making integration seamless.



Partition Level Refresh:

Instead of rebuilding the entire materialized view when source data changes, Doris intelligently refreshes only affected partitions, dramatically reducing refresh overhead and ensuring data freshness.

Query Layer

User SQL queries, dashboards, reporting tools

Materialized View Layer

Pre-computed aggregations & transformations
Optimized columnar storage, indexes

Iceberg Data Lake Layer

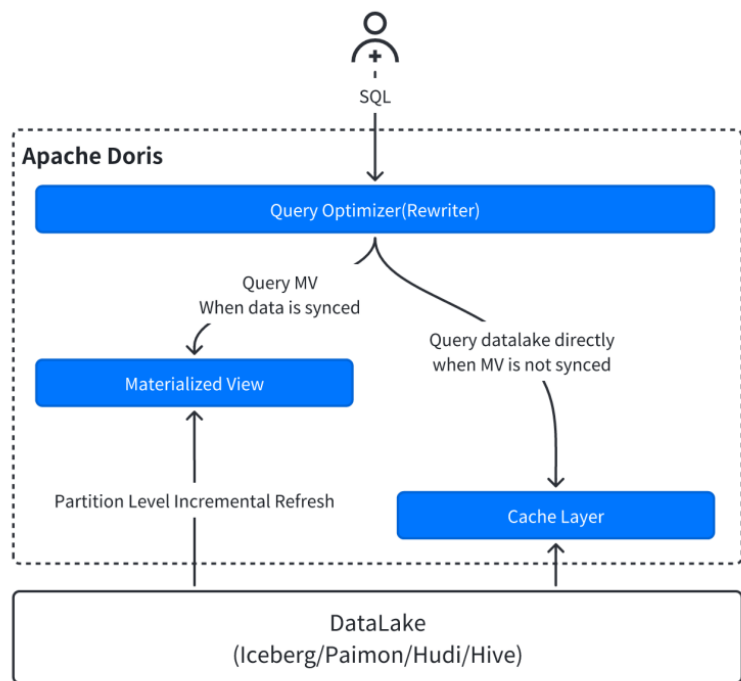
Raw data files, snapshots, manifests

10-100xFaster

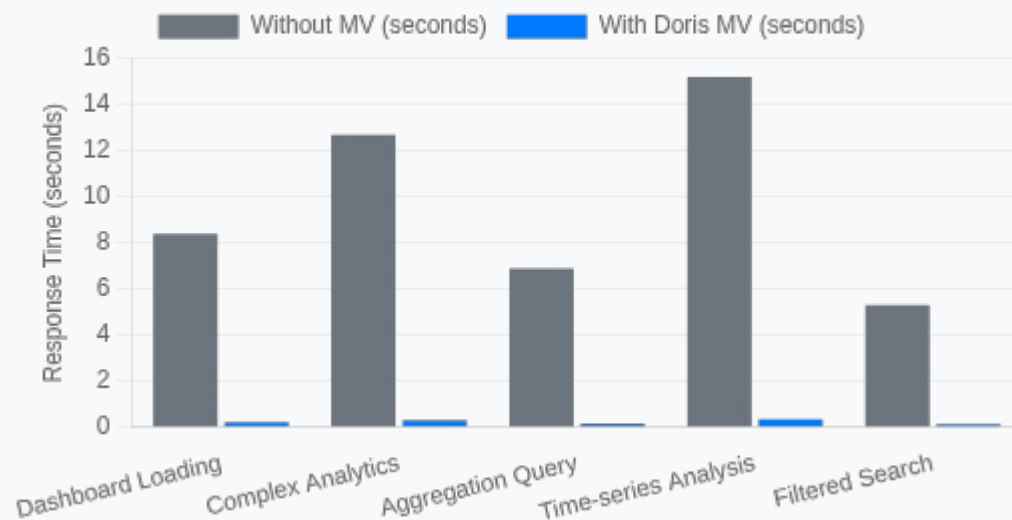


Materialized View: Showcase

Real-world case study: Large e-commerce analytics platform with 50TB+ of customer behavior data in Iceberg tables, experiencing performance issues with complex analytical queries across multiple dimensions.



Performance Comparison: Before vs. After Materialized Views



42×

Average query speedup

98%

Reduction in scan size

23 ms

Avg. dashboard response

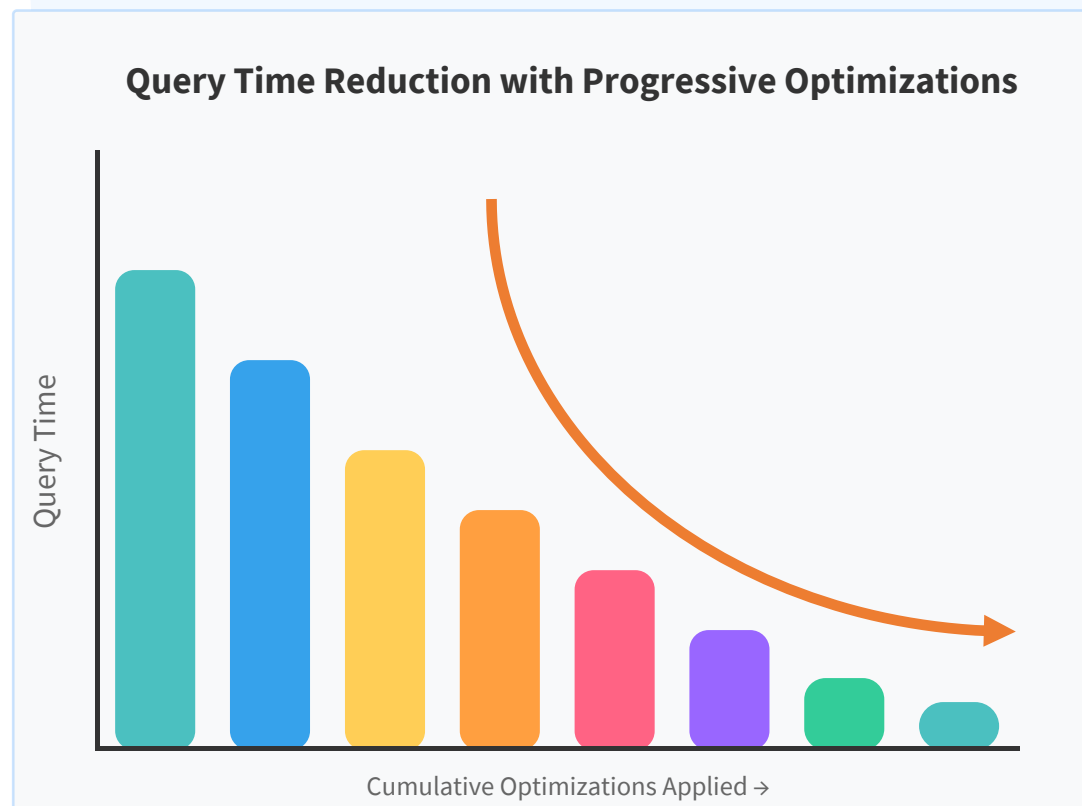
5x

Concurrent user capacity

Progressive Performance Optimization

Each optimization technique builds upon the previous ones, creating a compounding effect that dramatically reduces query time:

- | | |
|--|---|
|  Native C++ File Reader |  Lazy Materialized |
|  Merge IO |  Runtime Partition Pruning |
|  Prefetch |  Data Cache |
|  Predicate Pushdown |  More Optimization ... |



Beyond Lakehouse

What is Apache Doris and how it provides the “real-time” Capability

Real-Time Capability of Apache Doris(1)

Streaming Ingestion & Data Consistency

Apache Doris supports near real-time data ingestion through multiple mechanisms, ensuring data is available for analysis within seconds of generation while maintaining strong consistency guarantees.



Streaming Ingestion:

Native Kafka connector, Flink CDC integration, and stream load API enable second-level data ingestion from various sources with exactly-once semantics.



Multi-Table Consistency:

Transaction management across data sources ensures consistent snapshots, even with high-frequency updates from multiple streams.

Lakehouse Value:

These capabilities enable real-time analytics on streaming data while providing the same consistency guarantees expected from traditional databases - essential for modern lakehouse architectures.

Streaming Data Flow & Consistency Model

Streaming Sources

Kafka, Pulsar, CDC, IoT, Log Streams



Apache Doris Streaming Subsystem

Transaction Management, Exactly-Once, Low-Latency



Consistent Queryable Storage

ACID Guarantees, Real-Time Query Interface

Consistency Mechanisms:

- Transaction isolation ensures atomic updates across partitions
- Two-phase commit for distributed consistency guarantees
- Exactly-once semantics prevent duplicate ingestion
- Global catalog maintains coherent schema and metadata

Real-Time Capability of Apache Doris(2)

Update

Apache Doris has significantly evolved from a traditional append-only analytical database to a modern OLAP engine with comprehensive update capabilities, enabling new use cases beyond basic analytics.



Upsert Support:

INSERT, UPDATE, DELETE operations with standard SQL syntax enables ELT workflows and change data capture (CDC) integration.



Partial Updates:

Update only specific columns without rewriting entire rows, drastically improving performance for wide tables and sparse updates.

New Possibilities:

These capabilities bridge the gap between operational and analytical workloads, enabling real-time dashboards with continuously updated data, change tracking, and more flexible data pipelines.

Append-Only

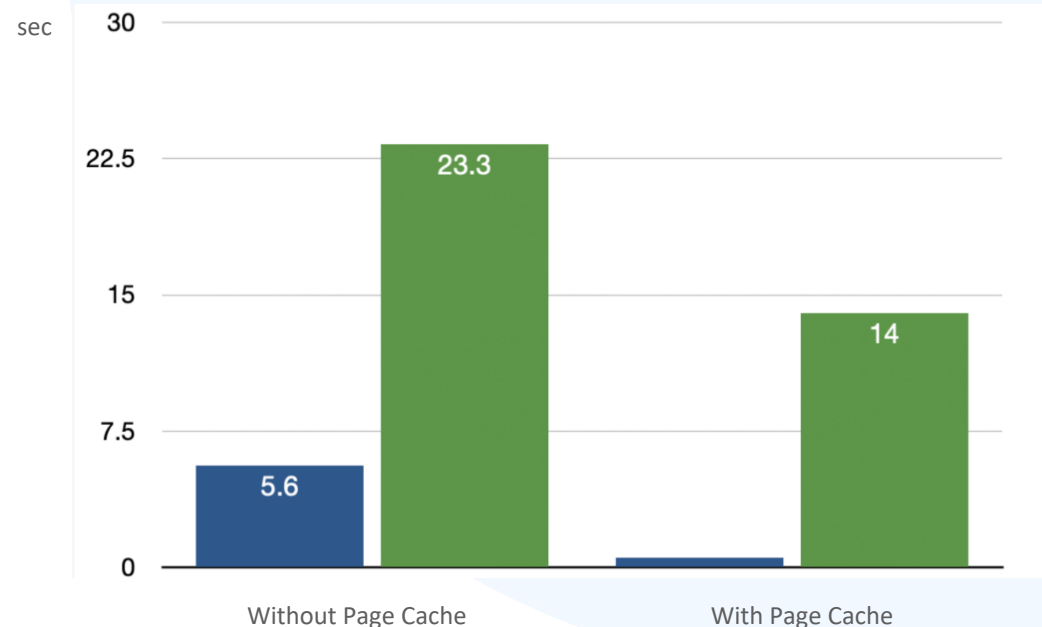
Basic INSERT operations

Basic Updates

Full row replacements

Advanced Updates

Partial updates & transactions



Query time in heavy update scenario
(Star-Schema Benchmark SF 100)

Real-Time Capability of Apache Doris(3)

High Performance

Apache Doris delivers exceptional query performance through its modern architecture designed specifically for real-time analytics at scale. The performance advantages come from several key technology innovations:



Vectorized MPP Engine:

SIMD-accelerated columnar execution with advanced data flow optimization enables processing billions of rows per second.



High Concurrency:

Multi-threaded query processing with intelligent resource management handles hundreds of concurrent users and queries.



Sub-second Latency:

Sophisticated query optimization, intelligent caching, and index technologies deliver consistent sub-second response times.

~ 1s

Minimum
data latency

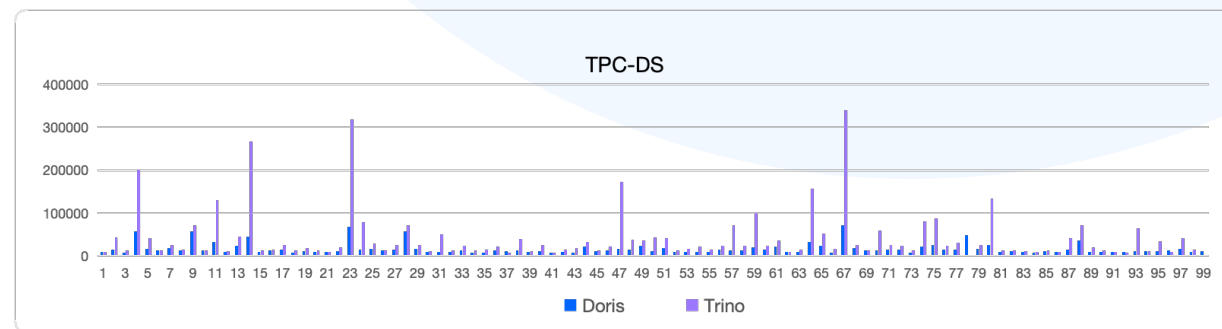
< 100ms

Average
query latency

> 10000 QPS

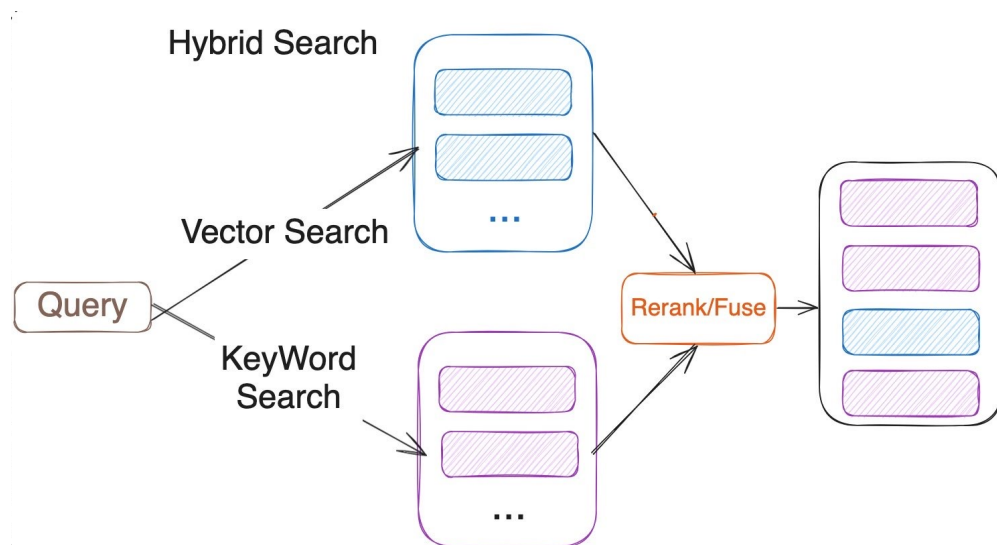
Maximum
Query Concurrency

System & Machine	Relative time (lower is better)
Umbra (c6a.metal, 500gb gp2):	x 1.61
ClickHouse (tuned, memory) (c6a.metal, 500gb gp2):	x 1.95
ClickHouse (tuned) (c6a.metal, 500gb gp2):	x 2.04
Apache Doris (c6a.metal, 500gb gp2):	x 2.15
ClickHouse (c6a.metal, 500gb gp2):	x 2.21
StarRocks (c6a.metal, 500gb gp2):	x 2.38
Umbra (c6a.4xlarge 500gb gp2):	x 2.40



Real-Time Hybrid Search for AI

Vector Search, Full-text Search, Structured Search



```

SELECT
  id,
  name,
  description
FROM docs
WHERE
  update_time BETWEEN '2025-01-01' AND '2025-05-30'
  AND description MATCH 'GenAI'
  AND version >= 2.0
ORDER BY
  cosine_distance(embedding, [...])
LIMIT 100
  
```

Why Vector-Capable General-Purpose Databases are Better for Enterprise GenAI ?

- Hybrid Query Capabilities
- Integration with Structured Data
- Simplified Operational Complexity and lower Costs
- Large-scale data volume, eg PBs

Looking Ahead: The Roadmap

What's Already Here: Doris + Iceberg Today

Apache Doris already provides robust integration with Iceberg, delivering a stable foundation for analytics on lake data. Organizations can immediately leverage these capabilities for real-world use cases without waiting for future enhancements.



Production-Ready Integration:

Doris's Iceberg connector is enterprise-tested and deployed across numerous production environments.



Versatile Data Operations:

Full support for read/write operations, time travel to historical data snapshots, and branch/tag navigation.

Use Cases You Can Build Today:

Real-time analytics dashboards on massive lake datasets

BI query acceleration without data duplication

Hybrid analytics joining lake data with operational data

Historical analysis using time travel capabilities

Data Read/Write Operations

Stable

Full support for efficient reads.
Append only insert

100%

Catalog Integration

Stable

Glue / HMS / Unity / Polaris / Gravitino / ...

100%

Time Travel

Stable

Query any historical snapshot with
simple syntax

100%

Branch & Tag Support

Stable

Read/write to branches, query
specific tags

100%

What's Coming Next: Doris + Iceberg Tomorrow

Building upon our current stable integration, the roadmap for Apache Doris and Iceberg brings enhanced capabilities to provide a complete lakehouse solution. Here's what we're working on next:



Fully Managed Iceberg Tables

In Progress

Complete lifecycle management of Iceberg tables directly from Apache Doris, including data rewrite operations and advanced snapshot management capabilities.



Iceberg V3 Support

In Progress

Deletion vector, row lineage, Variant, ...



Incremental Materialized Views

Planned

Efficiently refresh materialized views based on delta changes in source Iceberg tables, dramatically reducing refresh time and resource usage.

Ultimate Vision:

A complete lakehouse platform where Apache Doris provides the performance, governance and ease-of-use of a database while transparently leveraging Iceberg's data lake capabilities.

Community & Commercial Support

Welcome to Apache Doris Community

Apache Doris is a thriving open-source community with contributors and users worldwide. Join us to shape the future of this powerful MPP database and analytics engine. There are many ways to get involved:



Resources

Documents: <https://doris.apache.org/docs/gettingStarted/what-is-apache-doris>

Blog: <https://www.velodb.io/blogs>

EBooks & Slides: <https://www.velodb.io/ebooks-slides>



Contribution

Github: <https://github.com/apache/doris>

Guide: <https://doris.apache.org/community/how-to-contribute/contribute-to-doris>

Compilation: <https://doris.apache.org/community/source-install/compilation-with-docker>

Testing: <https://doris.apache.org/community/developer-guide/regression-testing>



Mailing List

<https://doris.apache.org/community/subscribe-mail-list>



Slack

apachedoriscommunity.slack.com

Linkedin:

<https://www.linkedin.com/company/velodb>

<https://www.linkedin.com/company/doris-apache/>

Youtube: https://www.youtube.com/@VeloDB_IO

Medium: <https://medium.com/@ApacheDoris>

X: https://x.com/doris_apache

About VeloDB

VeloDB.io

Enterprise-Grade Apache Doris Solutions

VeloDB provides enterprise-grade products and services built around Apache Doris, helping organizations maximize their data infrastructure investments with professional support and commercial enhancements.

Our Mission:

Bridge the gap between open-source innovation and enterprise readiness with production-grade Apache Doris solutions for both cloud and on-premise environments.

Managed Cloud Offerings

VeloDB Cloud: Fully managed, real-time data warehouse service. Support both SaaS and BYOC.

VeloDB Enterprise

Self-managed software on premises, on VMs, or K8s